10. Wayne

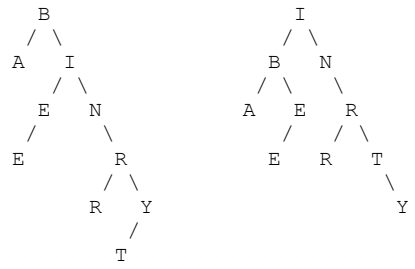
Program Name: Wayne.java **Input File: wayne.dat**

After completing a study unit on binary search trees, Wayne was considering how to keep these search trees better balanced. He had learned that the IPL (internal path length) of a tree is a good indicator of how balanced it is. He recalled that the **internal path length** of a tree is the sum of the depths of each node in the tree, and the better balanced a tree is, the lower the IPL value is. His idea was to take a data set, build a normal binary search tree by starting with the first element as the root node, and inserting the others in order, allowing duplicate values and **sliding ties to the left**. He would calculate the IPL of the normal tree, and then use a "two-ahead" technique in an attempt to build a more balanced BST (binary search tree).

This "two-ahead" process works like this. Consider the first three elements of the data set, arrange them in order, and insert the middle element of those three as the root. After that, add another element from the data set to the two remaining elements, arrange them in order again, and insert the middle element of those three. Continue this process until the last data element is added to the group, and again, insert the middle of those last three. When only two elements remain, insert the lesser of those two, and then insert the last one standing.

Supposedly, Wayne thought, this would make for a more balanced tree, which in turn would be more efficient. He decided to research this idea with several sets of data, consisting of capitalized words with no symbols or spaces, calculating the IPL of the standard BST for each word, and then the IPL of the "two-ahead" BST, and then determining if there was any improvement, and by how much. The IPL of the "improved" tree should be a lower value, but not always, as his research discovered. Here is how the two trees would look, using the characters in the word: **BINARYTREE**

Standard BST "Two-ahead" BST



The internal path length of the standard tree is 25 (two nodes 'A' and 'I' at depth 1, two at depth 2, two at 3, two at 4, and 1 at 5). The root node has a depth of zero, and does not affect the IPL. The "two-ahead" tree has an IPL of 21 (the two nodes 'B' and 'N' at depth 1, three at 2, three at 3, and one at 4). The difference shows a better IPL value by 4, which indicates a more balanced, and in turn, more efficient binary search tree. His report for the word "BINARYTREE" would be "`25 21 4 BETTER`", indicating the IPL for the standard tree, then the IPL for the "two-ahead" tree, the positive difference, and the word word `BETTER`, since 21 shows a more balanced tree.

Write a program to simulate Wayne's research, according to the process described above.

Input - Several words, all in caps, containing no symbols or spaces, each word at least three characters in length.

Output - For each word, build a standard binary search tree, with duplicate values allowed, and ties going to the left, and then output the IPL for that tree. Do the same using the "two-ahead" BST as described above and output the IPL for that tree. Finally, output the analysis, indicating the value and word that indicate how much "BETTER" or "WORSE" the second tree is from the first. All output items for one line will have single space separation.

Sample data:

```
BINARYTREE
RESEARCH
TWOAHEAD
```

Sample Output:

```
25 21 4 BETTER
16 16 SAME
19 21 2 WORSE
```